Expressiveness and Query Complexity in an Electronic Health Record Data Model

Robert H. Dolin, MD
Kaiser Permanente, Southern California Region
Robert.Dolin@kp.org

Objective: *Describe a high-level conceptual electronic health record (EHR) data model, explain how the model is expressive, present an algorithm for querying the model, and determine the complexity of this algorithm.* ***Design:*** *Entity-Relationship diagramming is used to represent the model, which relies on variably nested relations to enable expressiveness. The algorithm complexity is described using "big-oh" or "O( )" notation.* ***Results:*** *The data model appears to be highly expressive. A tractable recursive query processing algorithm is presented which is polynomial in time and space complexity.* ***Conclusion:*** *Several hurdles remain before the model and algorithm described can be fully tested in a live setting, including the development of techniques to populate the model. However, the study does show the ability to formally analyze an EHR model to understand its particular expressiveness and query complexity.*

INTRODUCTION

There is a wealth of information typically locked in narrative clinical reports unavailable to the computer for processing. Free text notes may contain complex concepts and relationships of varying depth and degree of interrelatedness[1]. Some of these data include symptom descriptions, discussions and weighting of diagnoses under consideration, and explanations of specific treatment plans. Whether these textual notes are ultimately fully parsable by natural language processing[2] or can be successfully entered using structured data entry[3] remains to be seen, but either way, the underlying data model must be capable of representing the concepts and relationships expressed by providers. However as a data model becomes more expressive, the same fundamental tradeoff between expressiveness and tractability encountered in other areas of medical informatics and computer science becomes relevant[4]. This paper describes the expressiveness of a previously reported data model[5-7]. A tractable algorithm for querying the model will be presented.

DEFINITIONS

Figure 1 uses Entity-Relationship (E-R) diagramming[8] to represent the conceptual EHR data model. In E-R modeling, entities (such as EHR_Patients) have attributes (such as Lastname or DOB) and relationships to one another (such as EHR_Patients having Many EHR_Concepts).

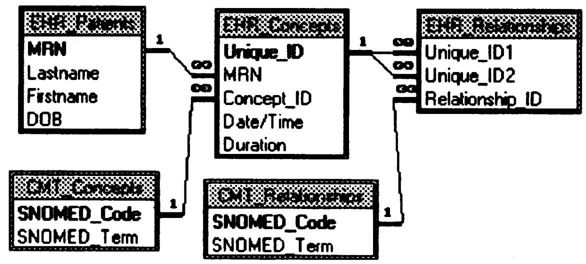


Figure 1. Conceptual data model.

The algorithm used to query this model will be analyzed for time complexity (i.e. how long it takes to complete) and space complexity (i.e. how much RAM and/or disk is required), expressed in "big-oh" or "$O(\ )$" notation[9]. If the running time $T(n)$ of some algorithm is $O(n^2)$ for example, then there are positive constants c and n_0 such that for n equal to or greater than n_0, $T(n) <= cn^2$.

DATA MODEL EXPRESSIVENESS

The expressiveness of a data model can be loosely defined as the extent to which it can represent, in an unambiguous codified format, those concepts and relationships expressed by providers, typically in their narrative free text reports. A basic data model would support the Subject-Property-Value paradigm (e.g. [nausea] -> (has-severity) -> [mild]). Such a model might be constructed of a Subject entity with a one-to-many relationship to a Properties entity. Evidence suggests that a model of this type (in the limited domains studied to date) might accommodate 80 to 90 percent of expressed concepts[3,10]. On the other hand, providers also wish to express comparisons with prior concepts or more complex relationships between findings[3,10].

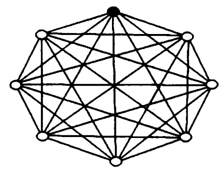
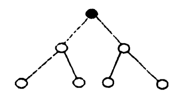
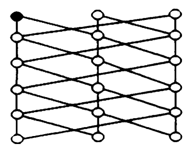
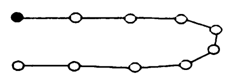
STRUCTURE	PICTURE	f(C)*	f(1000)
FULLY INTERCONNECTED		2	2
MONOHIERARCHY		Log_2C	10
CYLINDER		C / Base	100 (for Base=10)
LIST		C	1000

Figure 2. Sample data structures supported by data model in Figure 1.
* f(C) shows the number of iterations through the main loop of the algorithm in Figure 3, expressed as a function of C, which is the number of concepts in EHR_Concepts that are related to one another in EHR_Relationships.

Friedman, et al[1] have defined requirements for a clinical database, including the need to accommodate data that is variably nested (such as urine cultures where one or more organism is grown, each with its own set of antibiotic sensitivities), and data that is highly interrelated (such as Renal Insufficiency due both to Diabetes and Hypertension, and related to the symptoms of Polyuria and Malaise). Barrows and Johnson[11] have presented a data model that enables the representation of clinical thinking via variably nested relations, allowing queries such as "Show the final diagnoses for those patients initially identified solely as Anemia". Dolin[6-7] has previously described the model in this report, which supports the ability to relate any concept to any other concept. Extensions allow for the expression of time points, temporal intervals, and temporal uncertainty[5]. The European Standardization Committee (CEN), in its Electronic Healthcare Record Architecture draft standard[12] defines the concept of Record Complexes, which are comprised of 1-to-many Record Items. A Record Complex can also contain other Record Complexes.

These models and requirements rely on recursively defined definitions or variably nested relations. The model in this report supports this functionality as well as the ability to represent Subject-Property-Value pairs, thus appears to be highly expressive.

DATA MODEL QUERY ALGORITHM

The model referred to in this report uses variably nested relations to provide expressiveness, and uses a recursive query processing algorithm to extract the concepts and relationships of interest. The algorithm is modified from the semi-naive evaluation with static filtering algorithm described by Date[13] and Bancilhon and Ramakrishnan[14]. Given a particular concept of interest (as defined in EHR_Concepts), the algorithm will return the transitive closure of all relationships involving that concept (as they are defined in EHR_Relationships).

Data in this model might be structured in many ways, as shown in Figure 2. Nodes correspond to EHR_Concepts while arcs correspond to EHR_Relationships. Thus, if the input to the algorithm is the blackened node in any of the four pictures shown in Figure 2, all of the lines in that same picture will be returned.

The algorithm is shown in Figure 3. The concept(s) of interest (the Unique_ID value(s) from EHR_Concept) are inserted into table QUERY in line 3. From there, a loop is entered which first identifies all relationships involving concepts in table QUERY, and then identifies all concepts in newly identified relationships. Line 5 subtracts previously identified relationships and line 9 subtracts previously identified concepts, thus avoiding infinite loops. The loop terminates when table QUERY is empty, and returns table TUPLES, which will contain the transitive closure of all rows of table EHR_Relationships that were related to the inputted concept(s). The number of iterations through the loop depends on the particular data structure, as shown in Figure 2. Column 3 shows the number of iterations expressed as a function of the number of concepts in EHR_Concepts that are related to one another in EHR_Relationships. For 1000 concepts, the total number of iterations is shown in column 4.

QUERY COMPLEXITY

The time it takes for the algorithm to complete is approximately the sum of the times required for each iteration of the loop:

$$\sum_{i=1}^{x} T(\text{Line 4-9})$$

where i equals the iteration number, x equals the total number of iterations, and T(Line 4-9) equals the time required to execute lines 4 through 9. Since the running time of line 4 equals or exceeds that of lines 5, 6, 7, or 8, we can simplify the equation using the rule of sums for $O(\,)$ analysis[9]:

$$\sum_{i=1}^{x} T(\text{Line 4}).$$

The precise time of executing line 4 is difficult to determine because it is dependent on the particular machine, SQL compiler and database. As an approximation, the time is proportional to the size of QUERY, changing the above equation to:

$$\sum_{i=1}^{x} QUERY.$$

The change in QUERY with each iteration is dependent on the data structure, as shown in Figure 2. For a monohierarchy, QUERY grows exponentially, and the total number of iterations equals $\log_2 C$, where C is the number of concepts in EHR_Concepts that are related to one another in EHR_Relationships. For a list, QUERY remains constant, while the total number of iterations equals C. But any particular concept will be queried for at most once, and every concept can potentially be related to the concept(s) of interest. Therefore:

```
1   Create Tables TUPLES, TMP
        Unique_ID1        Integer,
        Unique_ID2        Integer,
        Relationship_ID   Integer;

2   Create Tables QUERY, QUERIED
        Unique_ID         Integer;

3   {Query for concept(s) of interest.}
    Insert Into QUERY (Unique_ID)
    Values ('X');

Repeat
4   {Identify all relationships that involve concepts in QUERY.}
    TMP :=
        (Select  EHR_Relationships.*
         From    EHR_Relationships, QUERY
         Where   EHR_Relationships.Unique_ID1
                        = QUERY.Unique_ID
                 OR
                 EHR_Relationships.Unique_ID2
                        = QUERY.Unique_ID);
5   {Subtract those relationships already identified.}
    TMP := TMP - TUPLES;
6   {Add newly identified relationships to TUPLES.}
    TUPLES := TUPLES + TMP;
7   {Keep track of all identified concepts.}
    QUERIED := QUERIED + QUERY;
8   {Identify all concepts in the newly identified relationships.}
    QUERY :=
        (Select TMP.Unique_ID1 From TMP
         UNION
         Select TMP.Unique_ID2 From TMP);
9   {Subtract those concepts already identified.}
    QUERY := QUERY - QUERIED;
Until QUERY = Null;

10      Return TUPLES;
```

Figure 3. Transitive Closure Recursive Algorithm. Returns the transitive closure of all tuples in EHR_Relationships that are related to the concept(s) of interest in EHR_Concepts.

$$\sum_{i=1}^{x} QUERY <= C.$$

As discussed above, the exact time of executing line 4 is difficult to determine precisely. If we say that the time when the size of QUERY is one tuple is equal to Q, then the upper bounds on the time required for the algorithm to terminate becomes:

$$C*Q.$$

For large databases that are searched on indexed fields, Q may be approximately on the order of Log R, where R is the number of tuples in the table being searched. Q, being constant, is dropped, resulting in:

$$O(C).$$

The amount of space required to run the algorithm is $O(R)$, where R equals the number of tuples in table EHR_Relationships, since TUPLES may equal the size of R, but cannot exceed R.*

EXAMPLE

The following example, shown in Figure 4, will illustrate the use of the algorithm in Figure 3. Window "Free Text" shows text that has been entered directly or via structured data entry. The concepts and relationships of that text have been extracted and are represented in window "EHR_Relationships". An undirected graph depiction of the concepts and relationships is shown in window "Visual". The goal is to determine if exercise (in this case yardwork [Y]) (or alternatively LUE ache [L]) preceded the chest pain [CP]. Because the concepts [CP] and [Y] (or [L] and [Y]) are not directly associated with one another in a single tuple in EHR_Relationships, a standard SQL query would fail to indicate a relationship. The concept [Y] is at a variable distance from two other potential concepts of interest, [CP] and [L].

Answering this query is complicated, although tractable. The initial step (which is addressed in this report) involves the use of the algorithm in Figure 3 to extract those relevant tuples from EHR_Relationships. This is accomplished by inserting the Unique_ID of concept [CP] into line 3 of the algorithm. The final step involves making inferences on those extracted tuples. Tractable algorithms for making the temporal inferences needed to answer the query in this example have been defined[15].

CONCLUSION

This report has described a tractable algorithm for determining the transitive closure on variably nested relations, which have been found by the author and other researchers to enable increased expressiveness in medicine. Of practical concern however is how this algorithm would perform in a live environment. Just getting the data into the model represents a major hurdle. Natural language parsers and techniques for structured data entry may represent solutions. The actual degree of nesting that would be recorded given a suitable user-interface remains to be seen. The use of indexing, the physical clustering strategy of the data, and SQL query design will have a significant impact on algorithm performance. It may be that performance limitations will limit the use of this algorithm to off-line studies and outcome analyses.

While the electronic health record has been viewed by some as a 'gold-mine' of information, the fact remains that as the computer represents more complex information, the ability to analyze that information also gets increasingly complicated. CJ Date made the statement regarding data modeling that "the problem of finding the logical design that is incontestably the right one is a rather intractable problem"[13]. However, it should be possible to formally analyze an EHR model to understand its particular expressiveness and query complexity.

Acknowledgments

The author thanks Ron Dolin, PhD Candidate, UCSB Department of Computer Science; Keith Campbell MD, PhD Candidate, Stanford University School of Medicine and consultant to Kaiser Foundation Health Plan; and HunSu Kim, Senior Systems Programmer, Kaiser Permanente for their insightful comments.

*An unoptimized SQL compliler potentially has space requirements of $O(C*R)$. If the product operation in the From clause in line 4 is performed before the restriction operations in the Where clause, the number of tuples in the intermediate table, which equals the number in QUERY times the number in EHR_Relationships, can be as large as $C*R$, since the size of QUERY can approximate C.

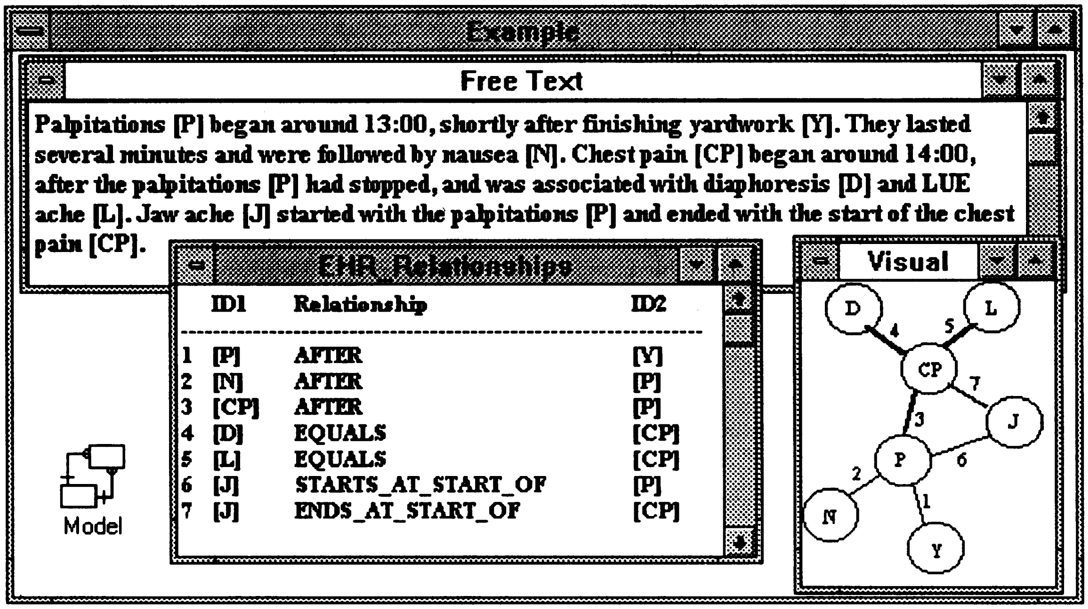


Figure 4. Example. See text for details.

References

1. Friedman C, Hripsack G, Johnson SB, Cimino JJ, Clayton PD. A generalized relational schema for an integrated clinical patient database. SCAMC 1990.
2. Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking clinical data from narrative reports: A study of natural language processing. Ann Intern Med 1995;122:681-88.
3. Moorman PW, van Ginneken AM, Siersema PD, van der Lei J, van Bemmel JH. Evaluation of reporting based on descriptional knowledge. JAMIA 1995;2:365-73.
4. Levesque HJ, Brachman RJ. Expressiveness and tractability in knowledge representation and reasoning. Comput Intell 1987;3:78-93.
5. Dolin RH. Modeling the temporal complexities of symptoms. JAMIA 1995;2:323-31.
6. Dolin RH. Modeling the relational complexities of symptoms. Methods Inf Med 1994;33(5):448-53.
7. Dolin RH. A high-level object-oriented model for representing relationships in an electronic medical record. SCAMC 1994:514-8.
8. Barker R. CASE Method - Entity Relationship Modeling. Reading MA: Addison-Wesle y 1990.
9. Aho AV, Hopcroft JE, Ullman JD. Data Structures and Algorithms. Reading, Mass: Addison-Wesley Publishing Company; 1983:1-427.
10. Bell DS, Greenes RA. Evaluation of UltraSTAR: performance of a collaborative structured data entry system. SCAMC 1994:216-22.
11. Barrows RC, Johnson SB. A data model that captures clinical reasoning about patient problems. SCAMC 1995:402-405.
12. CEN (European Committee for Standardization)/Technical Committee 251 - Medical Informatics. Project Team 011. Electronic Healthcare Record Architecture. Draft. CEN/TC251/PT011. June 6, 1995. (URL http://miginfo.rug.ac.be:8001/index.htm)
13. Date CJ. An Introduction to Database Systems, Volume I. Reading, Mass: Addison-Wesley Publishing Company; 1990:1-854.
14. Bancilhon F, Ramakrishnan R. An amateur's introduction to recursive query processing strategies. In: Stonebraker M, ed. Readings in Database Systems. San Mateo, CA: Morgan Kaufmann, 1988.
15. Esch JW, Nagle TW. Temporal intervals. In: Nagle TE, Nagle JA, Gerholz LL, Eklund PW, eds. Conceptual structures: Current research and practice. 1992:363-80.